EXPERIENCES IN BUILDING, OPERATING, AND USING THE ARPA NETWORK

David C. Walden

Bolt Beranek and Newman Inc.
Cambridge, Massachusetts

A retrospective written at the 5-year point

To be presented at the Second USA-Japan Computer Conference, Tokyo, Japan, August 1975.

EXPERIENCES IN BUILDING, OPERATING, AND USING THE ARPA NETWORK*

David C. Walden

Bolt Beranek and Newman Inc.
Cambridge, Massachusetts, USA

The author provides a retrospective look at the development and growth of the ARPA Network with emphasis on insights acquired through the experiences described.

1. INTRODUCTION

The ARPA Network development was begun in 1968 and 1969 [1,2,3,4,5]. By 1971 and 1972, the network was an operational entity, although considerable development has continued through the present date. Network growth has also continued, in the physical size of the network, in the uses of the network, and in traffic loads carried by the network. Within the next year it seems likely that responsibility for the ARPA Network will be transferred from ARPA to some other agency of the U.S. government, an agency better suited to the running of an operational (as opposed to an experimental) network. This would seem to be an appropriate moment, therefore, to reflect on the development of the network, its use, and some of the lessons learned.

We begin with a brief summary of the network's characteristics.

As shown in Figure 1, the ARPA Network has three distinct components: nodes, lines, and hosts. Connected to each node are from one to four independent computer systems (the hosts). The nodes themselves (called Interface Message Processors or IMPs) are connected together by leased phone lines, typically 50 Kbs circuits, although higher and lower speed lines are used in a few cases. Each node is connected to from two (or very occasionally one) to four (five are possible) other nodes. The nodes provide a communications subnetwork through which the hosts are able to communicate. A host wishing to communicate with another host presents messages (addressed entities less than about 8000 bits long -- similar to post office letters) to the node to which it is directly connected. The node receiving the message from a host breaks the message up into smaller entities (up to about 1000 bits long) called packets, for transmission from node to node across the network until the packets of the message all arrive at the node to which the destination host is connected. The packets of the message are then reassembled into the original message and delivered to the host. The individual packets are aided in their traversal of the network by a system of adaptive packet routing and a system of node to node packet retransmission when necessary.

As shown in Figure 2, the network currently spans the continental United States with overseas nodes in Hawaii, London, and outside Oslo. There are presently about fifty-five nodes in the network with about 100 hosts including twenty-five TIPs; the TIP [6] is a minimal host built into a node so as to allow terminals to be connected directly to the network without having to go through a separate host computer. Several additional nodes and hosts are presently scheduled for connection to the network.

These are, briefly, the characteristics and dimensions of the ARPA Network at present. We next consider in turn a) the growth of traffic carried by the network; b) the "technical results" of the network development; c) some present technical limits on further network growth and development; d) the operational lessons learned; e) some changes one might make if redoing the network; and f) the network's future.

*This work was sponsored by the Advanced Research Projects Agency of the U.S. Department of Defense under contracts DAHC-15-69-C-0179, F08606-73-C-0027, and F08606-75-C-0032.

2. TRAFFIC GROWTH

In early 1973, Roberts [7] presented a curve of average host internode traffic growth for the network which showed the level of internode network traffic to be increasing at a rate of a factor of ten every ten months. Internode traffic means traffic sent from a host on one node to a host on a different node (i.e., it does not include traffic sent between hosts on the same node); although Roberts does not consider intranode traffic, we will consider it at a later point in this paper. Based on this rapid rate of growth, Roberts predicted the network would run out of capacity in nine months. As shown in Figure 3, shortly after Roberts' prediction the rate of internode traffic growth decreased sharply to roughly a factor of two every twenty months. It is interesting to speculate on the reason for this sharp decrease in growth rate.

Before the network existed, ARPA apparently bought a computer for each of its research contractors. Once the existing computers were put on the network as hosts, ARPA has appeared more reluctant to provide each of its contractors with a host, preferring instead that new contractors use existing hosts; in fact, as documented in [8], some groups acquired all of their computing over the network. Thus, it can be hypothesized that the existing hosts (and the few new hosts that were added) were used remotely more and more and network traffic increased more and more, until the hosts (at least the popular time-sharing hosts) began to run out of capacity; this made it pointless for new remote users to attempt to get service, and resulted in turn in a leveling off of network traffic growth. Therefore, instead of the network running out of capacity as predicted by Roberts, it seems that the hosts ran out of capacity while the network still has capacity left. The hypothesis that the hosts have exhausted their capacity can presumably be tested by examining host load average records.

As already stated, the traffic shown in [7] and in Figure 3 includes only internode traffic. There are two reasons for excluding intranode traffic. First,

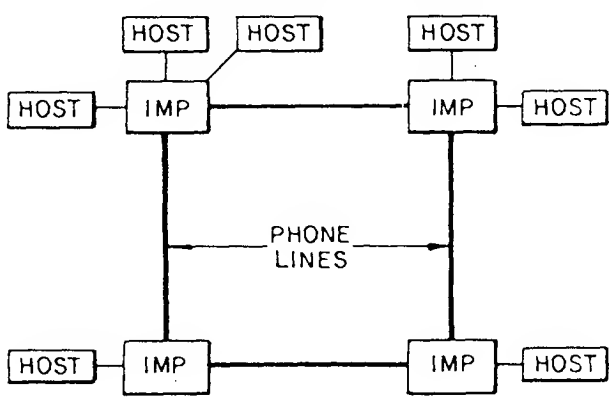

Figure 1. Nodes, lines, and hosts

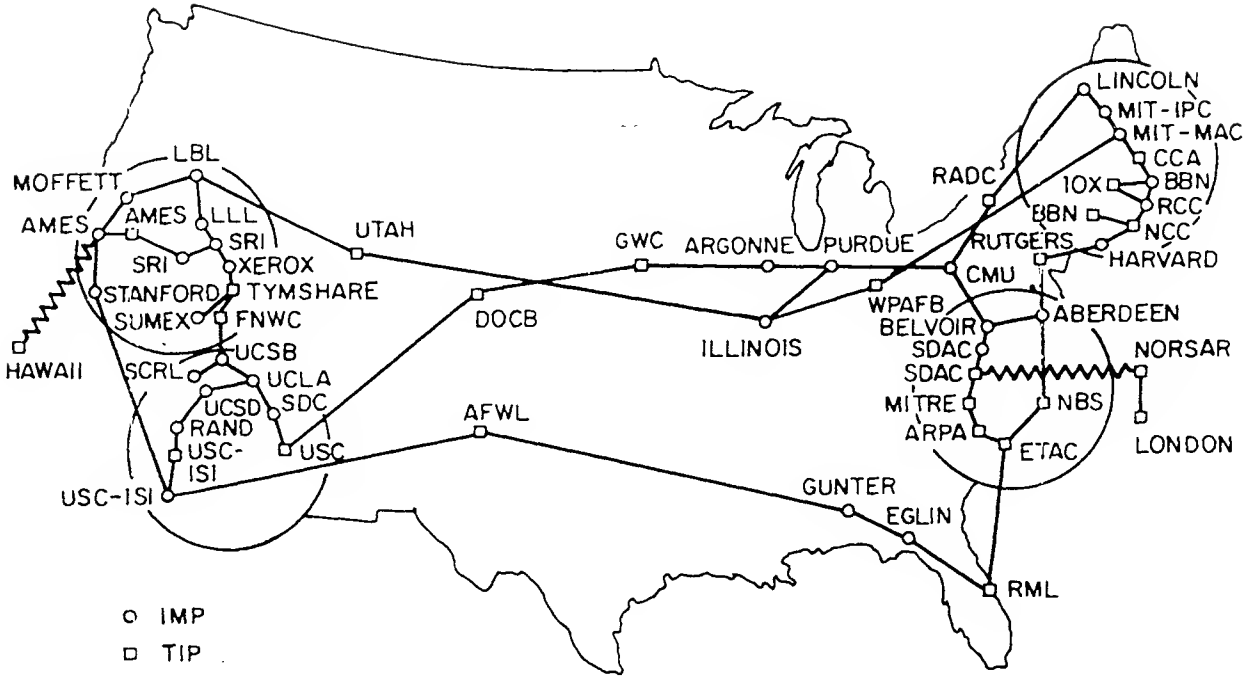

Figure 2. Geographic Map of the ARPA Network

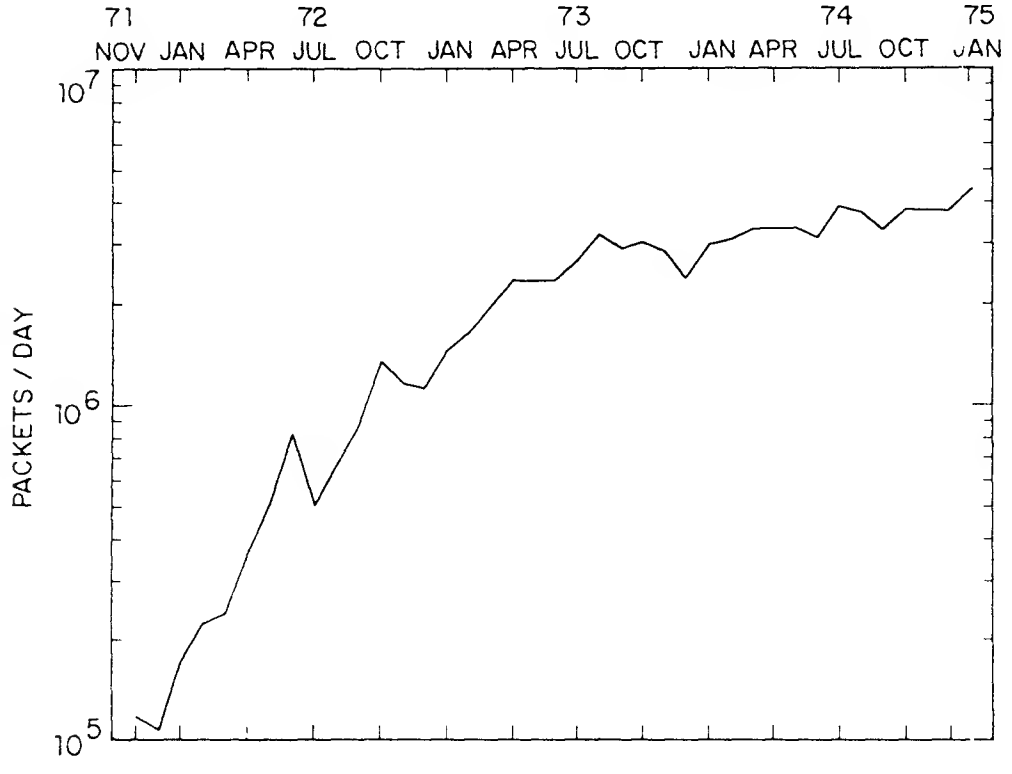

Figure 3. Growth in Average Host Intermode Traffic

intranode traffic puts a burden on only one node rather than on the network as a whole. Thus, when Roberts, for instance, was attempting to calculate the effects of host traffic on network capacity, he naturally excluded intranode traffic. Second, the available intranode traffic statistics include some amount of test traffic being looped from a host through its node and back to the same host, and there is no convenient way to separate this looped test traffic from actual data traffic between two hosts on the same node. It is believed, however, that there is actually a significant amount of real traffic between hosts on the same node. For instance, Kleinrock [9] reports that during a week-long measurement, the level of intranode traffic amounted to a daily average of twenty percent of the level of internode traffic, and in some one-hour intervals the intranode traffic level was as much as eighty percent of the internode traffic level. A scan of available long term statistics on inter- and intranode traffic shows that in recent months intranode traffic levels have averaged between twenty and forty percent of internode traffic levels. Thus, the traffic curve given in Figure 3 should be scaled up by this factor if all traffic is to be included.

That intranode traffic is a significant portion of all network traffic is interesting and probably indicative of four phenomena. First, the IMP is a handy interhost interface, and once one is installed in a computer center to connect some host onto the network, there is very soon pressure to connect other computers in the computer center to the IMP so that desired communication between the computers is possible. Second, when two computers are connected to the same IMP so that they may both communicate with other computers in the network, communication between the two computers themselves comes free and begins to happen even if it was not initially thought to be desired. Third, the TIP (a host) has been chosen by several sites as the most flexible available terminal multiplexor, and TIP-to-host traffic at these sites is likely to be intranode. Fourth, there is a (as yet still weak, but definite) tendency for hosts to be clustered at a certain site and therefore often on the same IMP. There are several reasons for this tendency: a) it is the inclination of some sites and computer center managers to accumulate machines; b) some sites need the service but have never wanted the burden of operating a computer center; c) some sites have shown particular proficiency at operating computer centers and a funding agency has ordered that machines be clustered at such sites; and d) there is an economy of scale possible when only one facility and staff is required for the operation of several computers. In any case, once the network became available, some sites have arranged with some other sites that one site's computer was moved to a second site, and the second site managed it for the first site which used the computer over the network via a simple terminal concentrator.

3. TECHNICAL RESULTS

One of the major initial goals of the ARPA Network development effort was to demonstrate the technical feasibility of packet-switching and several related concepts. Not all of the desired "proofs" are in yet. For instance, as mentioned in passing in the previous section, the network has not yet been subjected to severe loading of its throughput capacity. Thus, the capability of a packet-switching network to hold up under very heavy loads has not yet been operationally demonstrated, although there have been a number of experimental demonstrations of this capability. However, a number of significant technical results are in hand as a result of the ARPA Network development and operation, and we discuss several of the most important of these in the following paragraphs. Unfortunately, we do not have space to discuss all of the results: no doubt some of our colleagues in the ARPA Network development would have considered other results of equal or greater importance.

3.1 Adaptive Routing

The ARPA Network provides a convincing demonstration that adaptive routing algorithms can be made to perform reliably (e.g., in a globally correct manner in the face of local failures), efficiently (e.g., adapting to changes in the network quickly and accurately), and flexibly (e.g., accommodating a variety of circuit bandwidths and internode distances) without excessive complexity and overhead. The ARPA Network routing algorithm [10] was implemented, without particular difficulty and with modest code space, table space, and bandwidth requirement, right at the outset of the ARPA Network development. Since then it has run without fundamental change, although numerous improvements have been made. It has run every day and run well, with only a few failures per year; what failures have occurred have been related without exception to a bug in a change to the algorithm or to hardware failures (i.e., not because the algorithm had inherent instabilities), and it has been possible to prevent further occurrences of each detected failure with modest additional code. As argued in [11], we do not believe that the alternatives to adaptive routing for a distributed network (namely, fixed routing, random routing, and centralized routing) could have performed nearly so well in the growing, error-prone, operational environment of the ARPA Network.

3.2 Localized Failures

The ARPA Network has demonstrated that it is possible to build a large operational network in such a way that the effects of component failures are localized rather than "crashing" or otherwise making non-operational large portions or all of the network. A node or a host can fail in the ARPA Network and network use will be prevented for only the few users directly connected to that node or using that host. In the majority of cases when a line fails, no user is prevented from using the network. The network's ability to localize the effects of failures is based on a number of mechanisms, including: the provision of two or more network paths between most pairs of nodes; the nodes' ability to detect a line failure and to pass traffic queued for that line back for rerouting; the provision of an acknowledgment and retransmission mechanism on the lines to detect and recover from broken packets; the ability of the hosts to retransmit on the rare occasions when the nodes lose a message; and the adaptive routing algorithm. These mechanisms are described further in References 2 and 12.

3.3 Low Delays

The ARPA Network has confirmed the theoretical result [13] that networks which store-and-forward packets can achieve delays which are low when compared to the delays incurred in the computers (hosts) which are using the network. The ARPA Network was originally designed with the characteristic that a small message would take no more than .2 seconds to go from one host to another (this goal has recently been somewhat relaxed, primarily for economic reasons). A criticism of distributed networks and the ARPA Network in particular has been that .2 seconds is not particularly impressive compared with the .02 seconds it would take a short message to travel between two hosts if they were connected by a direct wire. However, in the ARPA Network we have found that delays on the order of .2 seconds are low enough that they are often hidden by the normal delays of going through the modern, complex operating systems of the hosts. Thus, it has been shown that the distributed network is not itself a delay-creating bottleneck.

3.4 Graceful Expansion and Contraction

The ARPA Network has demonstrated that a network can be constructed so that nodes, lines, traffic, and so on can be added or deleted without major upheavals with each addition. Proof of this capability is the fact that node and line addition decisions are made by groups other than the group responsible for the network software and hardware development and operation. Further, nodes (and occasionally lines) have been added on very short notice. For instance, a node was recently taken out of the network in Cleveland and then, on no more than a week's notice, shipped to Santa Barbara and installed without hitch (however, ARPA already had the necessary line). New users routinely come onto the network and add to its load without notification or arrangement with those responsible for development and operation. Contributing to the network's capability for graceful growth are the adaptive routing algorithm, the nodes' ability to multiplex packets of many host-to-host conversations onto the network lines, and the nodes' ability to dynamically allocate their storage and bandwidth to the tasks at hand.

3.5 Autonomous Control

The ARPA Network has demonstrated that it is possible for a network to control and operate itself without explicit control from a control center. Many network designs include at their heart the assumption that there must be a manned control center with the capa-

bility of adjusting the network's routing, instructing nodes which lines to use, adjusting nodal buffer storage, adjusting nodal priorities, and so forth. The ARPA Network depends on no such central control. All such adaptations are made by the nodes themselves, alone or in concert; further, these decisions are generally made more rapidly and accurately than could be done from a manned control center.

3.6 Central Monitoring, Maintenance, and Debugging

The ARPA Network has demonstrated new techniques for monitoring, maintenance, and debugging. Traditional networks have often required a staff at each node to perform these functions. The nodes in the ARPA Network typically operate at sites where there is no knowledgeable person available locally. The nodes automatically report their status (via the network itself) to a network monitoring center, and maintenance and debugging (of both the software and hardware) are typically carried out (again via the network) from the monitoring center. Of course, on some occasions it is necessary for a person to actually go to a node, for instance to replace a failed electronic component. Most interesting about these new centralized techniques is the fact that the maintenance and debugging is actually of a higher quality, because it is controlled by experts who can be concentrated at the center rather than by the less highly trained personnel normally available in field maintenance staffs. Of course, the field maintenance staff is still necessary when physical presence is needed at a field site; e.g., when the central personnel wish a hardware component changed.

3.7 Communication Between Heterogeneous Computers

Possibly the most difficult task undertaken in the development of the ARPA Network was the attempt -- which proved successful -- to make a number of independent host computer systems of varying manufacture, d varying operating systems within a single manufac- ured type, communicate with each other despite their diverse characteristics. For instance, in the ARPA Network an IBM 360/91 running OS [14] can successfully communicate with a PDP-10 running the TENEX [15] operating system; both can communicate with the MULTICS [16] operating system running on a GE645; and so on.

3.8 The Results in Perspective

It is not claimed that the ARPA Network method of host-to-host communications, and the other techniques discussed in this section, are the best techniques possible. Indeed, it would be surprising at this early stage in the development of computer networks if the best answer had been found in any of these areas. However, it is certain that in each of these areas a solution has been found that is feasible and that demonstrably works in the ARPA Network. The previously worrisome possibility that there might be no adequate solutions in these areas is layed to rest, and future network designers can use such techniques without fear of failure.

4. LIMITS

Because the ARPA Network was originally conceived as a research and development network, and because of historical accidents in its development, lack of farsightedness by its developers and so on, the network presently contains a number of limits to its capacity, flexibility, and reliability. Some of these limits are discussed below.

4.1 Node Memory, Bandwidth, and Fan-in Limits

The nodes in the network are based on the Honeywell 516 and 316 line of computers, the original members of which were designed around the mid-1960's. Thus, the nodes are using what is fundamentally a 10-year-old computer. As such, the computer has a limited memory address space, limited capability to attach numbers of I/O devices, and limited bandwidth. These limits mean at, for example, the nodes can have no more than ven devices (e.g., 3 line interfaces and 4 host interfaces) attached to a single node, there is no capability to add memory for packet buffering sufficient to allow the network to grow any larger (in terms of path length, not in terms of number of nodes) than it presently is, and the number and speed of terminals which can be attached to a TIP is restricted.

To break through this set of limits, ARPA has sponsored the development of a new line of computers to be used in a new, higher performance node. This line, called the Pluribus [17], utilizes a very general bus structure to enable construction of multi-resource (e.g., multi-processor, multi-memory, multi-I/O-bus) systems which can grow flexibly.

4.2 Network Topology

Initially the ARPA Network topology was specified to hold to the constraint that there be no more than seven nodes on the shortest path between two hosts wishing to communicate. Further, the topology was constrained to have sufficient capacity so that it was unlikely that serious queueing delays would be incurred between any two hosts wishing to communicate. It was claimed in [1] that these goals could continue to be met as the network developed to a size of 40 to 60 nodes with the cost of the lines growing only linearly with the size of the network.

In recent years, however, nodes (adding hops) and hosts (adding traffic) have been put on the network without addition of sufficient lines to continue to meet the initial delay goals. Today, therefore, if the current connectivity is maintained, the network cannot grow any larger without inserting intolerable delays for the users of the network. Further, the long strings of only doubly connected nodes pose a problem of reliability.

The solution to the present limiting topology is straightforward: more lines must be added to the network.

4.3 Reliability

There are reliability limits on the network in areas other than that touched on briefly in the previous paragraph. Because it was initially only a research and development network, the purpose of which was to demonstrate the feasibility of a new technology but not necessarily to provide perfect around-the-clock service, the ARPA Network is missing many of the features that would be mandatory in any communications utility. For instance, the nodes have no backup power source; thus, there are frequent node failures due to power failures. It has been estimated that merely by providing 10 minutes of standby power (i.e., a rechargeable battery), eighty-five percent of the node failures due to power failures could be avoided.

The nodes themselves have no backup. While the mean time to repair a failed node is only a few hours, it happens all too frequently that nodes are down for many hours and occasionally for days without replacement. The conventional solution to this problem is to supply an additional node at each site which can provide cold or hot stand-by for the primary node. ARPA, however, has attacked this problem again through the development of the Pluribus-based node, which can be configured with sufficient redundancy and capability for isolation of failed components so that when a node component fails the node continues to operate, albeit at reduced capacity, until the failed component can be repaired [18].

4.4 The Host/Node Interface and Host Performance

As presently implemented, the software interface between the nodes and their hosts is limiting in several ways. First, the interface, especially on the host side, contains insufficient instrumentation to allow quick and convenient determination of whether the node or the host is at fault in various failure situations. For instance, suppose a user complains that, when using a host across the network from a terminal on a nearby node, he sees excessive delays. It often takes days of simultaneous programmer investigation of all of the systems involved (the nodes, the host, and the terminal handler) to understand the source of the problem (if, indeed, the source is ever ascertained). Second, the host/node interface in most cases provides insufficient error control. Third, traffic going from a host to its node destined for a responsive host may be interfered with (i.e., delayed) by traffic from the host to a less responsive host. These three problems together cause a bearable but annoying limit on the quality of the host connection to the network, and consequently on the user's desire to use the network. These problems all spring from insufficient understanding at the time the interface was designed. There is a clear need for instrumentation of the host/node interface to facilitate further understanding.

Contributing to this limit is the fact that many of the hosts in the network cannot, when normally or heavily loaded, maintain "decent" throughput levels to the network. We believe the latter problem has three causes: a) the conventions used for communication between the hosts are excessively complex; b) most host operating systems were not designed with a network in mind and could not be made to accommodate a network connection effectively, i.e., communication with the network consumes an inordinate part of the hosts' capacity; and c) the host operating systems are generally suboptimal from the outset and the network interface just adds to the confusion (i.e., there are fundamental operating system problems to be overcome in addition to problems of the network interface).

The solution to the above depends on rethinking what the host/node interface should be in light of our experience, support of standard network interfaces as part of operating systems by computer manufacturers, and generally better operating system design by computer manufacturers. That is, the original ARPA Network approach is feasible (it has demonstrated that this interface can be designed) but it is certainly not optimal in terms of performance, flexibility, and so on.

4.5 Addressing and Routing

The node was initially specified to have only one host attached and the network was specified to have a maximum of about nineteen nodes. Thus the address fields in the node-to-node packet format, in the host-to-host message format, and in the host-to-node message format, which provide for four hosts attached to a node and sixty-three nodes in the network, will soon be a serious bottleneck if the network is to keep growing. It is a straightforward but major software change in all the nodes and hosts to expand the address fields so that they in turn allow expansion. If this change is to be made, the addresses should be expanded to allow at least a 16-bit node address and at least an 8-bit host address. (It is possible to make such a change in a backward-compatible manner.)

Once there may be more than sixty-three nodes in the network, the present one-level routing algorithm will become a bottleneck. Unless the routing algorithm is changed to operate in a manner using hierarchical (or area) routing, the routing calculation will begin to take excessive node and line bandwidth and node storage. Methods are being studied which provide adaptive, hierarchical routing.

5. OPERATIONAL LESSONS

As the network has grown and evolved, we have learned a number of lessons about operation of a distributed computer network. Some of these lessons we suspected from the beginning and our suspicions were confirmed very early. Other lessons we learned later and often with great difficulty. We feel it will be useful to list some of our insights here briefly. This topic is discussed further in Reference 19.

5.1 Loopback

In a system with as many components (i.e., hosts, lines, nodes) as the ARPA Network, it is crucial to have a loopback capability on all interfaces between components for the purpose of fault isolation. If this is to be conveniently possible, the interfaces must be symmetric (both hardware and software) and loop control must be possible from a remote location (e.g., the network monitoring center).

5.2 Software Distribution

In a large distributed system, it is impossible to distribute new software by sending tapes to each site for local loading. Further, it is almost impossible to make major changes to the software in a single step. A mechanism must be provided for remote loading (from the network development center) of software, and new software releases should be done in frequent small steps (e.g., it is better and will cause less disruption to change the software in small ways every week than to leave the software stable for many weeks and then attempt a single larger change, though this does take a node down for a few minutes every week). It is possible to envision some software changes which are not amenable to stepwise installation, but nonetheless stepwise installation is the goal to be sought.

5.3 Checksums

For a distributed system to operate correctly, all of the constituent parts (i.e., the nodes) must be operating correctly or they must be prevented from operating at all. Checksumming code, data structures, and data has proved to be a powerful technique for detecting both hardware and software failures and preventing the spread of a problem, although to date this technique has been used primarily at the node level rather than the host level.

5.4 Central Responsibility for All Components

The user of a distributed network is incapable of, and should not be expected to, distinguish between component failures. The user thinks of the whole network system (hosts, nodes, and lines) as if it were a single monolithic entity. When the user cannot use the system, whether because his terminal is broken, because the host he is trying to use is heavily loaded, or because a necessary node is down, he sees it as the network not working. Thus the network operators must somehow obtain the ability to control and trouble-shoot all components (including terminals and hosts), since it is the network operators who will receive the bulk of user complaints.

In the same area, as we mentioned in a previous section, central monitoring and maintenance of both hardware and software are superior to on-site monitoring and maintenance.

5.5 Quality Assurance

This is not a new lesson: it has been learned many times before in other applications. However, in a distributed system, where it is crucial that every component be functioning correctly, a strong and effective quality assurance program is all the more important. We believe that this means that hardware and software must be certified through extensive tests by individuals other than those responsible for development of software or for fabrication of the hardware. It would be preferable that those who will be responsible for maintenance of the software and the hardware should be able to veto the release of any new system until they are satisfied that they will be able to maintain that system.

5.6 Pressure

Finally, and again this is probably not a new lesson, we have found that there is unbearable user pressure for expansion of the network -- expansion of the variety of options available, expansion of the size of the network, expansion of the traffic levels, and so on. Such continuing expansion means that the network cannot be treated as a stable system which can be operated by competent technicians. Rather, the availability of a reasonably large staff of engineers, programmers, and analysts will be required for network operation. In our experience, it is not yet possible to develop a distributed computer network that can be installed on a turn-key basis.

6. CHANGES

It is fun (and sometimes enlightening) to ask the question, "What would you do differently if you were to do it all over again?" In this section we suggest a few of the things we might try to do differently if we were to begin again to build the ARPA Network. These speculations are all in the context of knowing from the beginning that the network would end up a communications utility. If the network were to remain a research tool, a different set of changes might be in order.

6.1 Central Offices

For reasons of increased control of the nodes, increased physical accessibility of the nodes, amortizing (bigger) nodes over more hosts, and amortizing backup equipment over more nodes, we would choose to build modular nodes which would reside in "central offices".

6.2 Large Memory Address Space

We believe that the single most constraining item in the development of the ARPA Network has been the available memory address space in the node computers.

Beginning again, we would not choose any computer which limited the memory address space and consequently the memory that can be used to as little as 32 kilowords of sixteen-bit memory.

6.3 Expanding the Network Functions

In the interests of improved fault diagnosis and increased maintainability, we would move more of the functions presently outside the control of the network operators to within their control. For instance, data sets and terminal access arrangements connected to the nodes would not be owned by other institutions or individuals, but would be owned by the network, maintained as part of the network, and made available as part of the network service. Similarly, "service bureau hosts" on the network, used by large numbers of users, would be maintained and operated by the network and made available as part of the network service. For those hosts which are not part of the network service, as much as possible of the host-to-host communication mechanism would be supplied by the network. Notice that this area of change runs counter to the current trend in much of the packet-switching community, which is to remove functions from the subnetwork of nodes and force them on the hosts; e.g. References 20, 21, and 22.

7. THE FUTURE

Only ARPA can accurately predict the future of the ARPA Network. Nonetheless, we feel we cannot conclude this paper of reflection on the network's construction, operation, and use without some comment on what the future may bring.

We have already mentioned that within the next year it is likely that management responsibility for the network will be transferred from ARPA to some more suitable government agency. Both before and after the transition, and in fact until commercial and government network utilities are widely available in two to five years' time, we believe that the ARPA Network will continue to grow and to provide service and a development test bed. Such growth will require lifting of the various limits mentioned earlier in this paper, and we believe this will be done. By virtue of its present rather large size, its head start in development, and its expected continued growth and development, we expect that the ARPA Network will continue to be (by far) the largest and the most advanced of the first generation of packet-switching networks. We look forward with interest to the second generation of packet-switching networks such as Telenet [23] and Autodin II [24].

ACKNOWLEDGMENTS

Too many individuals at too many institutions have participated in the construction, operation, and use of the ARPA Network for it to be feasible to list them here. Nonetheless, these individuals and institutions are deserving of acknowledgment for the portions of their work that have been reported in the present paper, and the author extends his gratitude to them collectively. The staff of the Information Processing Techniques Office of the U.S. Defense Department's Advanced Research Projects Agency, who conceived of and supported the ARPA Network development, deserve special credit. Bob Brooks, Alex McKenzie, and John McQuillan of Bolt Beranek and Newman Inc. were most helpful in advising the author on the form and content of this paper. The author is also specifically grateful to two members of the staff of the University of Southern California's Information Sciences Institute, Keith Uncapher who suggested that this paper be written and John Heafner who reviewed the paper.

REFERENCES

[1] L.G. Roberts and B.D. Wessler, "Computer Network Development to Achieve Resource Sharing," AFIPS Conference Proceedings, Vol. 36, June 1970, pp. 543-549.

[2] F.E. Heart, R.E. Kahn, S.M. Ornstein, W.R. Crowther, and D.C. Walden, "The Interface Message Processor for the ARPA Computer Network," AFIPS Conference Proceedings, Vol. 36, June 1970, pp. 551-567.

[3] L. Kleinrock, "Analytic and Simulation Methods in Computer Network Design," AFIPS Conference Proceedings, Vol. 36, June 1970, pp. 569-579.

[4] H. Frank, I.T. Frisch, and W. Chou, "Topological Considerations in the Design of the ARPA Computer Network," AFIPS Conference Proceedings, Vol. 36, June 1970, pp. 581-585.

[5] S.D. Crocker, J.F. Heafner, R.M. Metcalfe, and J.B. Postel, "Function-oriented Protocols for the ARPA Computer Network," AFIPS Conference Proceedings, Vol. 40, May 1972, pp. 271-279.

[6] S.M. Ornstein, F.E. Heart, W.R. Crowther, H.K. Rising, S.B. Russell, and A. Michel, "The Terminal IMP for the ARPA Computer Network," AFIPS Conference Proceedings, Vol. 40, May 1972, pp. 243-254.

[7] L.G. Roberts, "Network Rationale: A 5-Year Reevaluation," Proceedings COMPCON 1973, February 1973, pp. 3-6.

[8] M.S. Sher, "A Case Study in Networking," Datamation, Volume 20, No. 3, March 1974, pp. 56-59.

[9] L. Kleinrock and W. Naylor, "On Measured Behavior of the ARPA Network," AFIPS Conference Proceedings, Vol. 43, May 1974, pp. 767-780.

[10] J.M. McQuillan, "Adaptive Routing Algorithms for Distributed Computer Networks," BBN Report No. 2831, May 1974, available from the National Technical Information Service, AD781467.

[11] W.R. Crowther, F.E. Heart, A.A. McKenzie, J.M. McQuillan, and D.C. Walden, "Issues in Packet-Switching Network Design," to be presented at the AFIPS 1975 National Computer Conference, May 1975.

[12] J.M. McQuillan, W.R. Crowther, B.P. Cosell, D.C. Walden, and F.E. Heart, "Improvements in the Design and Performance of the ARPA Network," AFIPS Conference Proceedings, Vol. 41, December 1972, pp. 741-754.

[13] L. Kleinrock, "Communications Nets: Stochastic Message Flow and Delay," Dover Publications, New York, 1964, 209pp.

[14] International Business Machines Corporation, "The Functional Structure of OS/360," IBM Systems Journal, Vol. 5, No. 1, 1966.

[15] D.G. Bobrow, J.D. Burchfiel, D.L. Murphy, and R.S. Tomlinson, "TENEX, A Paged Time-sharing System for the PDP-10," Communications of the ACM, Vol. 15, No. 3, March 1972, pp. 135-143.

[16] E.I. Organick, "The MULTICS System: An Examination of Its Structure," M.I.T. Press, 1972.

[17] F.E. Heart, S.M. Ornstein, W.R. Crowther, and W.B. Barker, "A New Minicomputer/Multiprocessor for the ARPA Network," AFIPS Conference Proceedings, Vol. 42, June 1973, pp. 529-537.

[18] S.M. Ornstein, W.R. Crowther, M.F. Kraley, R.D. Bressler, A. Michel, and F.E. Heart, "Pluribus -- A Reliable Multiprocessor," to be presented at the 1975 National Computer Conference, May 1975.

[19] A.A. McKenzie, "The ARPA Network Control Center," January 1973, submitted for publication.

[20] Trans-Canada Telephone System, "Datapac Standard Network Access Protocol," November 1974.

[21] L. Pouzin, "Presentation and Major Design Aspects of the Cyclades Computer Network," Proceedings of the Third ACM Data Communications Symposium, November 1973, pp. 80-88.

[22] V. Cerf, "An Assessment of ARPANET Protocols," Proceedings of The 2nd Jerusalem Conference on Information Technology: Computers for Social and Economic Development, July 29 to August 1, 1974, pp. 653-664.

[23] Telenet Communications Corporation, "Before the Federal Communications Commission: Application for a Public Packet Switched Data Communications Network," October 9, 1973.

[24] R.D. Rosner, "A Digital Data Network Concept for the Defense Communications System," Proceedings of the National Telecommunications Conference, Atlanta, November 1973, pp. 22C1-6.